SHUMAKER & SIEFFERT, P.A.
1625 RADIO DRIVE, SUITE 300
WOODBURY, MINNESOTA 55125
TEL 651.735-1100
FAX 651.735-1102
WWW.SSIPLAW.COM

FACSIMILE SUBMISSION UNDER 37 CFR 1.8

TO: Mail Stop Appeal Brief - Patents	FROM: Kelly Patrick Fitzgerald
COMPANY: USPTO	DATE: JUNE 11, 2007
FAX NUMBER: 571-273-8300	TOTAL NO. OF PAGES INCLUDING COVER: 32
PHONE NUMBER:	SENDER'S REFERENCE NUMBER: 1014-013US01/JNP-0082
RE: Appeal Brief	APPLICATION SERIAL NUMBER: 10/045,717

RECEIVED
CENTRAL FAX CENTER
JUN 1 1 2007

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant/ Appellant:	Kireeti Kompella	Confirmation No.	9695
Serial No.:	10/045,717		
Filed:	October 19, 2001	Customer No.:	28863
Examiner:	Peling Andy Shaw		
Group Art Unit:	2144		
Docket No.:	1014-013US01/JNP-0082		
Title:	NETWORK ROUTING USING INDIRECT NEXT HOP DATA		

CERTIFICATE UNDER 37 CFR 1.8 I hereby certify that this correspondence is being transmitted via facsimile to the United States Patent and Trademark Office on June 11, 2007.

By: Caryl Harriman
Name: Caryl K. Harriman

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450,
Alexandria, VA 22313

Dear Sir:

This is an Appeal Brief responsive to the final Office Action mailed January 12, 2007. The Notice of Appeal was filed on April 11, 2007. Please charge Deposit Account No. 50-1778 the amount of $500, as required by 37 C.F.R. §41.37(a)(2) for a large entity. Please also charge any additional fees that may be required or credit any overpayment to Deposit Account No. 17-0026.

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party in interest is Juniper Networks, Incorporated, of Sunnyvale, California.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

STATUS OF CLAIMS

Claims 1-37 and 39-44 are on appeal in this case.

Claims 1-2, 5-6, 12-14, 16-18, 22-23, 28-31, 37 and 40 stand rejected under 35 U.S.C. 102(e) as being anticipated by Aramaki et al. (USPN 6,618,760) (hereafter Aramaki).

Claims 3-4, 7-11, 15, 19-21, 24-27, 32-36, 39 and 41-44 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Aramaki in view of Cain (USPN 6,857,026).

STATUS OF AMENDMENTS

The Application was originally filed on October 19, 2001 with claims 1-44.

The original claims were rejected by the Examiner in the Office Action mailed August 24, 2005.

Appellant amended claim 12 in a response filed on November 23, 2005.

The Examiner issued a final Office Action on February 23, 2006.

Appellant conducted a telephonic Examiner Interview on March 21, 2006.

Appellant then filed an Amendment with a Request for Continued Examination (RCE) on April 24, 2006, in which Appellant amended claims 1-3, 5-7, 9, 12-13, 17, 22-25, 28-30, 32-37, 39-41 and 43, and cancelled claim 38.

The Examiner issued an Office Action on July 7, 2006, which advanced the current rejections addressed in this Appeal.

Appellant filed a Response on October 6, 2006 without amending any claims.

The Examiner issued a final Office Action on January 12, 2007.

The claims on Appeal are those submitted in the Amendment filed on April 24, 2006.

RECEIVED
CENTRAL FAX CENTER
JUN 1 1 2007

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 recites a method comprising storing, within a network router, a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network[1]; storing, external to the forwarding tree, next hop data representing network devices neighboring the network router[2]; storing, within the leaf nodes of the forwarding tree, indirect next hop data that map the leaf nodes of the forwarding tree to the next hop data, wherein at least two different ones of the leaf nodes of the forwarding tree contain indirect next hop data that references the next hop data for the same neighboring network device[3]; identifying a key within a network packet[4]; traversing a subset of the nodes of the forwarding tree within a network device by testing at least one bit of the key per each of the traversed nodes, wherein values of the tested bits in the key determine a path traversed along the forwarding tree until reaching one of the leaf nodes of the forwarding tree[5]; upon reaching a leaf node of the traversed path, using the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree[6]; and forwarding the packet to the selected next hop[7].

Independent claim 12 recites computer-readable medium having data structures therein that control forwarding of packets by a network device[8] comprising a first data structure to store route data representing destinations within a computer network, wherein the first data structure is arranged as a forwarding tree having a set of nodes, and wherein the nodes includes a set of leaf nodes that correspond to destinations within a computer network[9]; a second data structure external to the forwarding tree to store next hop data representing interfaces to neighboring network devices[10]; and a set of data structures,

---

[1] See FIG. 2A and paragraph [0026].
[2] See FIG. 2A and paragraphs [0026] and [0028].
[3] See FIG. 2A and paragraphs [0028] and [0030].
[4] See paragraph [0027].
[5] See paragraph [0027].
[6] See paragraph [0028].
[7] See paragraph [0029].
[8] See paragraph [0012].
[9] See paragraph [0026].
[10] See paragraph [0028].

within the leaf nodes of the forwarding tree, to store indirect next hop data that map the leaf nodes of the forwarding tree to the next hop data[11], wherein the indirect next hop data causes the network device to, upon reaching a leaf node of a traversed path through the forwarding tree, select a next hop from the next hop data external to the forwarding tree and forward the packet to the selected next hop[12].

Independent claim 17 recites a router comprising a computer-readable medium[13] to store: (i) a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network[14], and (ii) next hop data, external to the forwarding tree, representing neighboring network devices[15], and (iii) indirect next hop data, within the leaf nodes of the forwarding tree, that maps the leaf nodes of the forwarding tree to the next hop data[16]; and a control unit[17] that identifies a key within a network packet, traverses a path through the forwarding tree by testing bits of the key until reaching one of the leaf nodes of the forwarding tree[18], wherein, upon reaching a leaf node of the traversed path, the control unit uses the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree[19] and forwards the packet to the selected next hop[20].

Independent claim 24 recites a router[21] comprising a routing engine to store routing information representing a topology of a network[22]; and a packet forwarding engine to store packet forwarding information in accordance with the routing information[23], the packet forwarding information including (i) a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network[24], and (ii) next hop data external to the forwarding tree, representing

[11] See paragraph [0028].
[12] See paragraphs [0028] and [0029].
[13] See paragraph [0011].
[14] See paragraph [0026].
[15] See paragraphs [0026] and [0028].
[16] See paragraph [0028].
[17] See paragraph [0038].
[18] See paragraph [0027]
[19] See paragraph [0028].
[20] See paragraph [0029].
[21] See FIG. 4.
[22] See paragraph [0039].
[23] See paragraph [0040].
[24] See paragraph [0026].

interfaces to neighboring network devices[25], and (iii) indirect next hop data, within the leaf nodes of the forwarding tree, that maps the leaf nodes of the forwarding tree to the next hop data[26].

Independent claim 28 recites computer-readable medium having instructions therein for causing a programmable processor within a router to store, within a network router, a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network[27]; store, external to the forwarding tree, next hop data representing network devices neighboring the network router[28]; store, within the leaf nodes of the forwarding tree, indirect next hop data that map the leaf nodes of the forwarding tree to the next hop data, wherein at least two of the leaf nodes of the forwarding tree contain indirect next hop data that references the next hop data for the same neighboring network device[29]; identify a key within a network packet[30]; traverse a subset of the nodes of the forwarding tree within a network device by testing at least one bit of the key per each of the traversed nodes, wherein values of the tested bits in the key determine a path traversed along the forwarding tree until reaching one of the leaf nodes of the forwarding tree[31]; upon reaching a leaf node of the traversed path, use the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree[32]; and forward the packet to the selected next hop[33].

Independent claim 37 recites a method comprising routing packets within a network using indirect next hop data that maps leaf nodes of a forwarding tree to next hop data stored external to the forwarding tree[34], wherein the leaf nodes correspond to destinations within a computer network[35], wherein the next hop data represents next hops

---

[25] See paragraph [0028].
[26] See paragraphs [0026] and [0028].
[27] See paragraph [0026].
[28] See paragraph [0028].
[29] See paragraphs [0026], [0028] and [0030].
[30] See paragraph [0027].
[31] See paragraph [0027].
[32] See paragraph [0028].
[33] See paragraph [0029].
[34] See paragraph [0028].
[35] See paragraph [0026].

within a network[36], and wherein at least two different ones of the leaf nodes of the forwarding tree contain indirect next hop data that references a same next hop within the next hop data[37].

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Appellant submits the following grounds of rejection to be reviewed on Appeal:

(1) The first ground of rejection to be reviewed is the rejection of claims 1-2, 5-6, 12-14, 16-18, 22-23, 28-31, 37 and 40 under 35 U.S.C. 102(e) as being anticipated by Aramaki.

(2) The second ground of rejection to be reviewed is the rejection of claims 3-4, 7-11, 15, 19-21, 24-27, 32-36, 39 and 41-44 under 35 U.S.C. 103(a) as being unpatentable over Aramaki in view of Cain.

ARGUMENT

Appellant respectfully traverses the current rejections advanced by the Examiner, and requests reversal by the Board of Patent Appeals based on the arguments below. The applied references fail to disclose or suggest features of Appellant's claims. Several different sets of claims are presented under separate headings. Accordingly, Appellant requests review of each of the different sets of claims presented under the separate headings.

FIRST GROUND OF REJECTION UNDER APPEAL

(Claims 1-2, 5-6, 12-14, 16-18, 22-23, 28-31, 37 and 40)

Claims 1-2, 5-6, 12-14, 16-18, 22-23, 28-31, 37 and 40 stand rejected under 35 U.S.C. 102(e) as being anticipated by Aramaki. In order to demonstrate anticipation under 35 U.S.C. 102, it is well established that the Examiner bears the burden of showing

[36] See paragraphs [0026] and [0028].
[37] See FIG. 2A and paragraph [0030].

that a prior art reference discloses each and every element of the claim in question. This well known rule of law is commonly referred to as the "all-elements rule."[38] If a prior art reference fails to disclose any element of a claim, then a rejection under 35 U.S.C. 102 is improper.[39]

In this case, Aramaki fails to disclose or suggest one or more features of Appellant's claims. Therefore, the anticipation rejections under 35 U.S.C. 102 are improper and must be reversed. More specifically, Aramaki lacks a suggestion of not one, but several features of Appellant's independent claims.

In the following discussion, Appellant focuses on the features recited in independent claim 1. Although the other independent claims do not recite the exact features of claim 1 or have identical scope, several arguments advanced below also apply with respect to the other independent claims. Indeed, the Examiner's analysis in the final Office Action focused on independent claim 1 and its dependent claims. For purposes of the arguments presented under this heading, claim 1 may be viewed as representative of the other independent claims. However, all independent claims do not necessarily stand or fall together insofar as additional arguments are presented below with respect to independent claim 24.[40]

Independent method claim 1 recites storing, within a network router, a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network. In addition, method claim 1 specifically requires next hop data external to the forwarding tree, where the next hop data represents network devices neighboring the network router. Claim 1 further recites storing, within the leaf nodes of the forwarding tree, indirect next hop data that map the leaf nodes of the forwarding tree to the next hop data that is external to the tree, wherein at least two different ones of the leaf nodes of the forwarding tree contain indirect next hop data that references the next hop data for the same neighboring network device. Thus, claim 1

[38] *See Hybritech Inc. v. Monoclonal Antibodies, Inc.*, 802 F.2d 1367, 231 USPQ 81 (CAFC 1986) ("it is axiomatic that for prior art to anticipate under 102 it has to meet every element of the claimed invention").
[39] *Id. See also Lewmar Marine, Inc. v. Barient, Inc.* 827 F.2d 744, 3 USPQ2d 1766 (CAFC 1987); *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (CAFC 1990); *C.R. Bard, Inc. v. MP Systems, Inc.*, 157 F.3d 1340, 48 USPQ2d 1225 (CAFC 1998); *Oney v. Ratliff*, 182 F.3d 893, 51 USPQ2d 1697 (CAFC 1999); *Apple Computer, Inc. v. Articulate Systems, Inc.*, 234 F.3d 14, 57 USPQ2d 1057 (CAFC 2000).
[40] Notably, independent claim 24 is addressed below under a separate heading.

requires that at least two of the leaf nodes contain indirect next hop data that map the two or more leaf nodes to the same next hop data external to the tree, thus mapping the two or more leaf nodes of the forwarding tree to the same next hop.

In addition, claim 1 recites identifying a key within a network packet, and traversing a subset of the nodes of the forwarding tree within a network device by testing at least one bit of the key per each of the traversed nodes, wherein values of the tested bits in the key determine a path traversed along the forwarding tree until reaching one of the leaf nodes of the forwarding tree.[41] Finally, claim 1 recites upon reaching a leaf node of the traversed path, using the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree, and forwarding the packet to the selected next hop.[42]

The concept of a forwarding tree can be understood with reference to the radix tree illustrated in FIG. 2A. of Appellant's specification, which is reproduced below.

[41] These specific features are also recited in most of the other independent claims, but are notably omitted from claim 37.
[42] These features are also not included in claim 37.

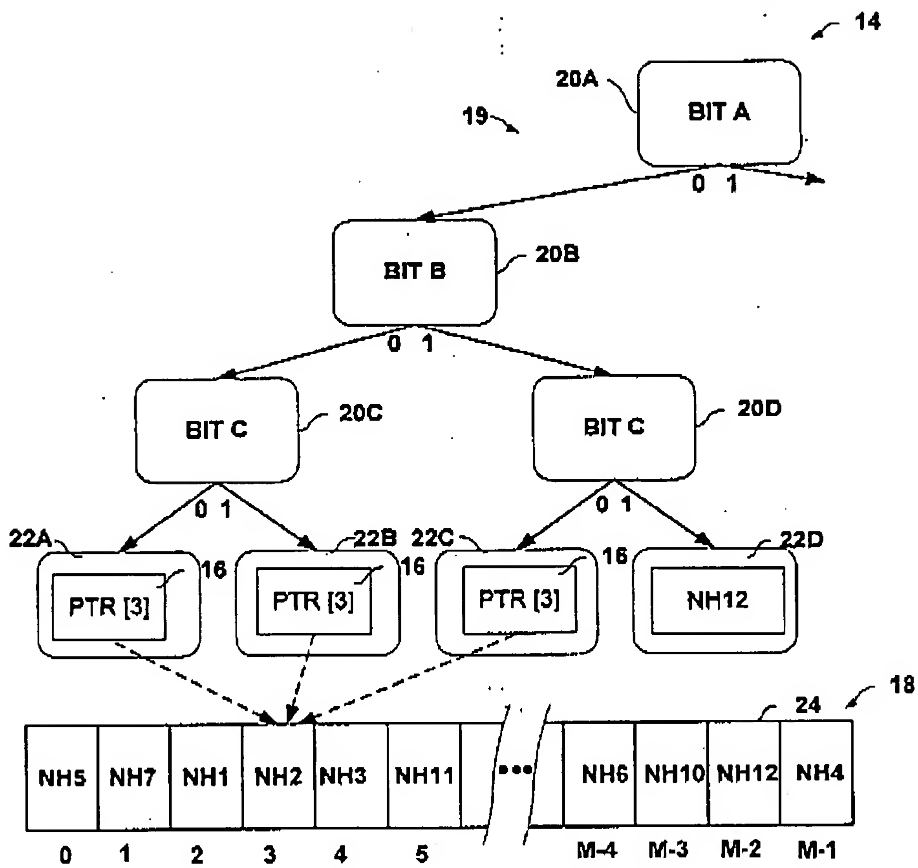

FIG. 2A

FIG. 2A illustrates an example of a forwarding tree in the form of a radix tree 19. As can be seen in FIG. 2A, the bits of a key (bits A, B and C) are tested to traverse the tree and define one of the leaf nodes 22. The different leaf nodes correspond to different destinations on the computer network, and define indirect next hop data 16 that maps to next hop data in a data structure 18 that is separate from radix tree 19. In particular, at least two of the leaf nodes 22A, 22B and 22C contain indirect next hop data 16 that map the two or more leaf nodes to the same next hop data (NH2) external to the tree 19, thus mapping the two or more leaf nodes of the forwarding tree 19 to the same next hop (NH2).

In the Office Action, the Examiner indicated that Aramaki discloses all of these features recited in claim 1 and rejected claim 1 (and the other independent claims) under 35 U.S.C. 102(e) as being anticipated by Aramaki. Contrary to the Examiner's analysis, however, several features of claim 1 (and similar features of various other claims) are not disclosed or suggested by Aramaki. Therefore, the rejections are improper and should be reversed.

Prior to discussing the specific passages cited by the Examiner from Aramaki, Appellant first notes that the teaching of Aramaki is not related to packet forwarding scheme that uses a forwarding tree (such as a radix tree) whatsoever, as recited in Appellant's claims. Instead, the teaching of Aramaki describes a packet forwarding scheme that uses sets of routing tables to be used instead of forwarding-tree based schemes. While Aramaki mentions radix tree retrieval methods in the Background section as relevant prior art for routing schemes, none of the features of Appellant's claims that recite particular unique features of a radix tree routing scheme or other type of forwarding tree routing scheme (addressed below) are suggested by Aramaki. Instead, Aramaki clearly distinguishes radix tree retrieval methods from his improved "expansion methods" that are based on retrieval tables as described in the Detailed Description of Aramaki. To be sure, the Background of Aramaki clearly distinguishes these different methods, first mentioning binary tree retrieval methods (col. 2, lines 5- 21), next explaining radix tree methods (col. 2, lines 22-51), and then explaining conventional expansion methods that use retrieval tables (beginning on col. 2, lines 52) that form the basis for the Aramaki invention.

The actual teaching in the Detailed Description of Aramaki concerns expansion methods that are purportedly different than the conventional table based expansion methods. The teaching of Aramaki in the Detailed Description does not appear to concern radix tree retrieval methods whatsoever.

The Examiner's analysis plucks disparate passages of Aramaki that discuss the different radix tree (in the background) and expansion methods (in the detailed description) and misconstrues these unrelated passages as teaching a system or technique not contemplated by Aramaki. When properly understood, the discussion of radix tree methods in Aramaki (in the Background) have no relevance or nexus with the table-based

techniques discussed in the Detailed Description of Aramaki. Furthermore, the forwarding techniques taught by Aramaki are different from those set forth in Appellant's claim 1.

Specifically, in the final Office Action, the Examiner stated that Aramaki discloses storing a forwarding tree having a set of nodes, and storing, external to the forwarding tree, next hop data representing network devices neighboring the network router. This statement, however, is incorrect. Nothing in Aramaki teaches or suggests storing a forwarding tree having a set of nodes, and storing, external to the forwarding tree, next hop data representing network devices neighboring the network router.

The Examiner's analysis cites many disparate passages of Aramaki. In the "Response to Arguments" section of the final Office Action, the Examiner appears to now be relying on column 5, lines 48-67 for these features of storing a forwarding tree having a set of nodes, and storing, external to the forwarding tree, next hop data. This passage of Aramaki from column 5, lines 48-67 is reproduced below:

> According to the present invention, an improved method for retrieving a hop pointer from a retrieval table using a destination address of a received data signal as a retrieval key to determine a next forwarding destination of the received data signal is provided. First, a plurality of first tables and a second table are stored in the retrieval table. The first tables are hierarchically arranged according to division of the destination address, an entry of each of the first tables including a second-table pointer indicating a next accessed second entry of the second table. The second table serves as an index table of the first tables, wherein each of entries of the second table includes a hop pointer and a first-table pointer indicating a next accessed first table. The second table and a selected one of the first tables are alternately accessed depending on a first-table pointer included in an accessed entry of the second table while retrieving an entry of an accessed first table using a corresponding divisional portion of the destination address. In this manner, a hop pointer is read from a finally accessed entry of the second table as a retrieval result.

It should be readily apparent that this teaching of Aramaki does not concern any use of forwarding tree, such as a radix tree like that shown in Appellant's FIG. 2A. Instead, the teaching of Aramaki makes use of two tables. The second table serves as an index table of the first tables. However, neither of these tables includes any forwarding tree. Instead, these tables are used as routing tables within a routing scheme based on the "expansion methods." As specifically stated by Aramaki's Background, such techniques do not use radix trees and the Examiner's statement to this effect is erroneous.

In the final Office Action, the Examiner specifically stated that "the first table is for storing a forwarding tree (citing column 1, lines 27-31 and column 5, lines 48-67). This statement, however, misconstrues Aramaki. Nothing in Aramaki suggests that the first table stores or otherwise represents a forwarding tree. Indeed, Aramaki specifically outlines the differences between conventional methods that use forwarding trees and expansion methods that use forwarding tables instead of forwarding trees. See the background of Aramaki.

The passage of Aramaki at column 1, lines 27-31 is part of the background in Aramaki and provides that:

> The router retrieves route information from the retrieval table by using an IP address indicating the ultimate destination of the incoming IP packet signal as a retrieval key, and determines a forwarding destination such as "output interface". In this retrieval, the router uses the longest

It should be apparent that the retrieval table discussed in column 1, lines 27-31 (like the first table discussed in column 5, lines 48-67) does not store any forwarding tree. Moreover, these different passages of Aramaki are entirely unrelated.

Immediately following the passage reproduced above, Aramaki goes on to describe, at column 1, lines 32-57, that the retrieval method using the retrieval table makes use of the "longest matching method." The longest matching method simply compares the full set of bit values of different entries of the table, and does not traverse any forwarding tree, such as shown in Appellant's FIG. 2A.

The data stored in the retrieval tables of Aramaki is, quite simply, not a forwarding tree. Along these lines, furthermore, the "longest matching method"

discussed in Aramaki with respect to table-based retrieval of destination addresses does not have any relevance to other features of claim 1, which further requires identifying a key within a network packet; traversing a subset of the nodes of the forwarding tree within a network device by testing at least one bit of the key per each of the traversed nodes, wherein values of the tested bits in the key determine a path traversed along the forwarding tree until reaching one of the leaf nodes of the forwarding tree; upon reaching a leaf node of the traversed path, using the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree; and forwarding the packet to the selected next hop.

Again, the only mention of forwarding trees in Aramaki appears to be the background discussion of binary tree retrieval methods, and radix tree retrieval methods. However, these methods are clearly distinguished by Aramaki relative to table-based approaches.

Therefore, it is entirely unclear why the Examiner would conclude that either of the tables discussed on column 5, lines 48-67 store a forwarding tree. Nothing in Aramaki suggests storing a forwarding tree having a set of nodes, and storing, external to the forwarding tree, next hop data representing network devices neighboring the network router, as required by Appellant's claim 1.

In addition, even if the teaching of Aramaki could be interpreted as teaching the storage of external next hop data, Aramiki does not do so in the context of a forwarding tree. The disparate, unrelated teachings in Aramaki regarding table-based routing techniques and alternative forwarding tree-based techniques do not, either alone or collectively, suggest storing a forwarding tree having a set of nodes, and storing, external to the forwarding tree, next hop date representing network devices neighboring the network router, as required by claim 1.

The Examiner's conclusions appear to even contradict the teaching of Aramaki, which expressly describes the use of retrieval tables as an alternative to (i.e., replacement of) the use of a forwarding tree. There is no suggestion in Aramaki to somehow use both forwarding trees and retrieval tables; nor is there any teaching as to how to link the structures in some hybrid fashion.

In the Office Action, the Examiner also stated that Aramaki discloses storing, within leaf nodes of the forwarding tree, indirect next hop data that maps the leaf nodes of the forwarding tree to the next hop data, wherein at least two different ones of the leaf nodes of the forwarding tree contain indirect next hop data that references the next hop data for the same neighboring network device. For this conclusion, the Examiner cited column 1, lines 27-31 and column 5, line 41 to column 6, line 3 of Aramaki.

These entire passages, however, are irrelevant to forwarding tree methods, and instead concern methods for retrieving hop pointers from retrieval tables. The passage at column 5, line 41 to column 6, line 3, for example, does not suggest storing indirect next hop data "within leaf nodes of a forwarding tree", as required by claim 1. This feature is, quite simply, lacking from Aramaki.

Again, the vague mention in Aramaki of conventional radix tree methods in the Background section, and the unrelated teaching in Aramaki of a packet forwarding scheme that instead uses routing tables having internal pointers would not have suggested, in any way, a technique in which indirect next hop data is stored within leaf nodes of a forwarding tree so as to map the leaf nodes of the forwarding tree to externally stored next hop data, as required by Appellant's claim 1.

The Examiner also stated, in the Office Action, that Aramaki discloses using the indirect next hop data within the leaf node of a traversed path of a forwarding tree (such as a radix tree) to select a next hop from the next hop data external to the forwarding tree. For this conclusion, the Examiner cited column 6, line 17 to column 7, line 10. This conclusion, however, has no factual basis. In particular, the passage at column 6, line 17 to column 7, line 10 fails to suggest indirect next hop data within a leaf node of a traversed path of a forwarding tree, and also fails to suggest the selection of a next hop from the next hop data external to the forwarding tree. Instead, this passage of Aramaki concerns a retrieval method that uses tables, and not a forwarding tree at all, much less a forwarding tree that stores indirect next hop data within its leaf nodes that map to next hop data external to the forwarding tree.

In summary, almost every feature of claim 1, when properly read in context of the literal language of the claim and the specification, is lacking from the Aramaki reference. Again, the teaching of Aramaki is not related to packet forwarding scheme that uses a

forwarding tree. Instead, the teaching of Aramaki describes a packet forwarding scheme (referred to as an expansion method) that uses sets of routing tables. While Aramaki mentions radix tree retrieval methods in the background section, Aramaki clearly distinguishes such radix tree retrieval methods from the expansion methods that use retrieval tables. Moreover, many of the unique features of a forwarding-tree and use of indirect next hop data within the leaf nodes of a forwarding tree, as recited in Appellant's claims, are not disclosed or event suggested by the Aramaki reference.

Appellant respectfully requests reversal of all pending rejections based on the discussion above.

Claim 24

In addition to the features addressed above, independent claim 24 also recites a specific router architecture. According to claim 24, a routing engine stores routing information representing a topology of a network, and a packet forwarding engine stores the route data, the indirect next hop data and the next hop data. These additional features of claim 24 are similar, e.g., to those recited in dependent method claim 8.

Notwithstanding the Examiner's anticipation rejection of claim 24, with regard to claim 8 (which recites similar features to claim 24), the Examiner recognized that Aramaki fails to disclose a routing engine stores routing information representing a topology of a network, and a packet forwarding engine stores the route data, the indirect next hop data and the next hop data. Thus, the anticipation rejection of claim 24 contradicts the Examiner's own analysis of claims 8, 33 and 42.

The anticipation rejection of claim 24 must be reversed for this additional reason. These features are addressed further in the discussion of claims 8, 33 and 42, below.

SECOND GROUND OF REJECTION UNDER APPEAL

(Claims 3-4, 7-11, 15, 19-21, 24-27, 32-36, 39 and 41-44)

Claims 3-4, 7-11, 15, 19-21, 24-27, 32-36, 39 and 41-44 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Aramaki in view of Cain. These rejections are improper for all the reasons advanced above with respect to the independent claims. In addition, Appellant requests separate review of various ones of the dependent claims

addressed under separate headings below. The rejections advanced for the various dependent claims addressed below are improper insofar as the secondary Cain reference fails to disclose or suggest the features attributed to this reference by the Examiner.

Claims 3 15 and 19

Dependent claims 3, 15, and 19 are dependent upon independent claims 1, 12 and 17 respectively. In addition to the arguments advanced above with respect to the independent claims, Appellant requests further review of these claims.

Dependent claim 3 requires storing, within each of the leaf nodes, a first reference to a primary next hop within the next hop data external to the forwarding tree, and storing, within each of the leaf nodes, a second reference to a backup next hop within the next hop data external to the forwarding tree. Dependent claims 15 and 19 require that data pointers (the indirect next hop data) include pointers to primary next hops and pointers to backup next hops.

In the Office Action, the Examiner noted that these features of claims 3, 15 and 19 are not disclosed or suggested in Aramaki. However, the Examiner cited Cain for these features and stated that a person of ordinary skill in the art would have been motivated to modify the teaching of Aramaki in view of Cain to arrive at Appellant's claimed invention.

Appellant submits, however, that the Examiner has misinterpreted Cain relative to the features of claims 3, 15 and 19. The cited passages of Cain discuss routing techniques that specify primary and backup *routes*. In contrast claims 3, 15 and 19 recite pointers to primary and backup *next hops*. The claim language, as well as Appellant's specification, makes clears that next hops from a routing device are different than routes through a network. A route specifies a beginning and an ending node, as well as every intermediate node along the route. A next hop, in contrast, is simply data representing a neighboring network device. Techniques that specify primary and backup routes, per Cain, would not have provided any suggestion for storing, within each of the leaf nodes, a first reference to a primary next hop within the next hop data external to the forwarding tree, and storing, within each of the leaf nodes, a second reference to a backup next hop within the next hop data external to the forwarding tree.

For this additional reason, Appellant requests reversal of the rejections of claims 3, 15 and 17.

Claims 7, 32 and 41

Claim 7 depends upon claim 1 and further recites receiving a packet comprising network update information, and modifying the next hop data external to the forwarding tree in response to the network update information without modifying the forwarding tree. Claims 32 and 41 recite similar features.

For these claims the Examiner noted that the additional features are not disclosed or suggested in Aramaki. However, the Examiner cited Cain for these features and stated that a person of ordinary skill in the art would have been motivated to modify the teaching of Aramaki in view of Cain to arrive at Appellant's claimed invention.

In particular, the Examiner cited a passage of Cain that discusses the reception of control messages at a router, followed by an update of network routes and route prioritizations. Appellant submits that Cain teaches the antithesis of claims 7, 32 and 41 insofar as Cain teaches modification of network routes. Modification of network routes (if a forwarding tree were used) would comprise modification of a forwarding tree rather than modification to next hop data without any modification of a forwarding tree, as recited in Appellant's claims 7, 32 and 41.

Nothing in Cain suggests receiving a packet comprising network update information, and modifying the next hop data external to the forwarding tree in response to the network update information without modifying the forwarding tree. On the contrary, in Cain, if a network update message is received, then the routes need to be recalculated. One specific advantage of Appellant's claimed invention is that such route recalculation can be avoided. In contrast to route recalculation, per Cain, claims 7, 32 and 41 recite modification to next hop data (which resides external to a forwarding tree) without any modification of the forwarding tree.

Claims 8, 33 and 42

Claims 8, 33 and 42 specifically require the use of a routing engine and a packet forwarding engine. For example, claim 8 recites storing routing information within a

routing engine, wherein the routing information represents routes within a network, and storing the route data, the indirect next hop data and the next hop data within a packet forwarding engine.

For these claims the Examiner noted that the additional features are not disclosed or suggested in Aramaki. However, the Examiner cited Cain for these features and stated that a person of ordinary skill in the art would have been motivated to modify the teaching of Aramaki in view of Cain to arrive at Appellant's claimed invention.

Specifically, the Examiner cited passages of Cain that discuss route computation logic and route logic that updates routing tables based on the computed routes. While the route logic and routing tables may be similar to a routing engine that stores routing in formation that represents routes within a network, nothing in these cited passages suggests anything akin to a packet forwarding engine, much less a packet forwarding engine that stores the route data, the indirect next hop data and the next hop data. The teaching of Cain appears to be a routing engine, and nothing more.

Appellant again notes that the routing engine and packet forwarding engine features (which the Examiner recognized as being lacking from Aramaki) are actually recited in independent claim 24. Accordingly, the anticipation rejection of claim 24 contradicts the Examiner's own analysis of claims 8, 33 and 42.

Claims 9, 25, 34 and 43

Claims 9, 25, 34 and 43 recite similar features to claims 8, 33 and 42. In addition, these claims require issuing a message from a routing engine to the packet forwarding engine to direct the packet forwarding engine to modify the next hop data. For example, claim 9 requires receiving a packet comprising network topology update information, updating the routing information within the routing engine, and
issuing a message from the routing engine to direct the packet forwarding engine to modify the next hop data in response to the network update information.

For these claims the Examiner again noted that the additional features are not disclosed or suggested in Aramaki. However, the Examiner cited Cain for these features and stated that a person of ordinary skill in the art would have been motivated to modify the teaching of Aramaki in view of Cain to arrive at Appellant's claimed invention.

However, as discussed with respect to claims 8, 33 and 42, Cain fails to disclose anything akin to a packet forwarding engine, much less a much less a packet forwarding engine that stores the route data, the indirect next hop data and the next hop data. Furthermore, contrary to the Examiner's analysis of claims 9, 25, 34 and 43 nothing in Cain suggests receiving a packet comprising network topology update information, updating the routing information within the routing engine, and
issuing a message from the routing engine to direct the packet forwarding engine to modify the next hop data in response to the network update information.

CONCLUSION OF ARGUMENT

The rejections of all pending independent claims should be reversed for the arguments presented with respect to the first grounds of rejection on Appeal. In addition, the rejection of independent claim 24 should be reversed for the additional reason presented with respect to the separate heading for that claim.

With respect to the second grounds of rejection on Appeal, Appellant has presented arguments for four different sets of claims under separate headings. To the Extent that the Board disagrees with Appellant's arguments for the first grounds of rejection on Appeal, Appellant respectfully requests separate review of each of the four different sets of claims that have been presented under separate headings with respect to the second grounds of rejection on Appeal.

Respectfully submitted,

Date: June 11, 2007

By:

Name: Kelly Patrick Fitzgerald
Reg. No.: 46,326

Shumaker and Sieffert
1625 Radio Drive, Suite 300
St. Paul, Minnesota 55125
Telephone:(651) 286-8343
Facsimile: (651) 735-1102

CLAIMS APPENDIX:

Claim 1 (Previously presented): A method comprising:

storing, within a network router, a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network;

storing, external to the forwarding tree, next hop data representing network devices neighboring the network router;

storing, within the leaf nodes of the forwarding tree, indirect next hop data that map the leaf nodes of the forwarding tree to the next hop data, wherein at least two different ones of the leaf nodes of the forwarding tree contain indirect next hop data that references the next hop data for the same neighboring network device;

identifying a key within a network packet;

traversing a subset of the nodes of the forwarding tree within a network device by testing at least one bit of the key per each of the traversed nodes, wherein values of the tested bits in the key determine a path traversed along the forwarding tree until reaching one of the leaf nodes of the forwarding tree;

upon reaching a leaf node of the traversed path, using the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree; and

forwarding the packet to the selected next hop.

Claim 2 (Previously presented): The method of claim 1, wherein the forwarding tree comprises a radix tree.

Claim 3 (Previously presented): The method of claim 2, wherein storing the indirect next hop data comprises:

storing, within each of the leaf nodes, a first reference to a primary next hop within the next hop data external to the forwarding tree, and

storing, within each of the leaf nodes, a second reference to a backup next hop within the next hop data external to the forwarding tree.

Claim 4 (Original): The method of claim 3, further comprising routing packets to the backup next hop in response to a network event.

Claim 5 (Previously presented): The method of claim 2, wherein storing the indirect next hop data comprises storing a data pointer within each of the leaf nodes that references the next hop data external to the forwarding tree.

Claim 6 (Previously presented): The method of claim 1, wherein storing the next hop data comprises storing an array of next hop data elements external to the forwarding tree.

Claim 7 (Previously presented): The method of claim 1, further comprising:

receiving a packet comprising network update information; and

modifying the next hop data external to the forwarding tree in response to the network update information without modifying the forwarding tree.

Claim 8 (Original): The method of claim 1, further comprising:

storing routing information within a routing engine, wherein the routing information represents routes within a network; and

storing the route data, the indirect next hop data and the next hop data within a packet forwarding engine.

Claim 9 (Previously presented): The method of claim 8, further comprising:

receiving a packet comprising network topology update information;

updating the routing information within the routing engine; and

issuing a message from the routing engine to direct the packet forwarding engine to modify the next hop data in response to the network update information.

Claim 10 (Original): The method of claim 8, wherein storing the routing information includes storing a copy of the route data, the indirect next hop data and the next hop data stored within the packet forwarding engine.

Claim 11 (Original): The method of claim 9, wherein storing the routing information includes storing a copy of the route data, the indirect next hop data and the next hop data stored within the packet forwarding engine, and issuing the message comprises analyzing the copy to identify the next hop for modification.

Claim 12 (Previously presented): A computer-readable medium having data structures therein that control forwarding of packets by a network device comprising:

a first data structure to store route data representing destinations within a computer network, wherein the first data structures is arranged as forwarding tree having a set of nodes, and wherein the nodes includes a set of leaf nodes that correspond to destinations within a computer network;

a second data structure external to the forwarding tree to store next hop data representing interfaces to neighboring network devices; and

a set of data structures, within the leaf nodes of the forwarding tree, to store indirect next hop data that map the leaf nodes of the forwarding tree to the next hop data,

wherein the indirect next hop data causes the network device to, upon reaching a leaf node of a traversed path through the forwarding tree, select a next hop from the next hop data external to the forwarding tree and forward the packet to the selected next hop.

Claim 13 (Previously presented): The computer-readable medium of claim 12, wherein the forwarding tree comprises a radix tree.

Claim 14 (Original): The computer-readable medium of claim 12, wherein the indirect next hop data comprises a set of data pointers stored within the leaf nodes.

Claim 15 (Original): The computer-readable medium of claim 14, wherein the data pointers include pointers to primary next hops and pointers to backup next hops.

Claim 16 (Previously presented): The computer-readable medium of claim 12, wherein the second data structure comprises an array of next hop data elements.

Claim 17 (Previously presented): A router comprising:

a computer-readable medium to store: (i) a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network, and , (ii) next hop data, external to the forwarding tree, representing neighboring network devices, and (iii) indirect next hop data, within the leaf nodes of the forwarding tree, that maps the leaf nodes of the forwarding tree to the next hop data; and

a control unit that identifies a key within a network packet, traverses a path through the forwarding tree by testing bits of the key until reaching one of the leaf nodes of the forwarding tree,

wherein, upon reaching a leaf node of the traversed path, the control unit uses the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree and forwards the packet to the selected next hop.

Claim 18 (Original): The router of claim 17, wherein the indirect next hop data comprises a set of data pointers stored within the leaf nodes.

Claim 19 (Original): The router of claim 18, wherein the data pointers include pointers to primary next hops and pointers to backup next hops.

Claim 20 (Original): The router of claim 17, wherein some of the next hop data represents software modules for processing data packets.

Claim 21 (Original): The router of claim 20, wherein each of the software modules is selected from one of a packet filter, a policy enforcer and a packet counter.

Claim 22 (Previously presented): The router of claim 17, wherein the forwarding tree is arranged as a radix tree.

Claim 23 (Previously presented): The router of claim 22,

wherein the indirect next hop data includes a set of data pointers associated with the leaf nodes, and

wherein the data pointers reference portions of the next hop data stored external to the forwarding tree.

Claim 24 (Previously presented): A router comprising:

a routing engine to store routing information representing a topology of a network; and

a packet forwarding engine to store packet forwarding information in accordance with the routing information, the packet forwarding information including (i) a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network, and , , (ii) next hop data external to the forwarding tree, representing interfaces to neighboring network devices, and (iii) indirect next hop data, within the leaf nodes of the forwarding tree, that maps the leaf nodes of the forwarding tree to the next hop data.

Claim 25 (Previously presented): The router of claim 24, wherein the routing engine receives a packet comprising network topology update information and, in response to the network topology update information, updates the routing information and directs the packet forwarding engine to modify the next hop data.

Claim 26 (Original): The router of claim 24, wherein the routing information includes data structures storing a copy of the route data, the indirect next hop data and the next hop data stored within the packet forwarding engine.

Claim 27 (Original): The router of claim 26, wherein the routine engine analyzes the data structures to identify the next hop for modification.

Claim 28 (Previously presented): A computer-readable medium having instruction therein for causing a programmable processor within a router to:

store, within a network router, a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network;

store, external to the forwarding tree, next hop data representing network devices neighboring the network router;

store, within the leaf nodes of the forwarding tree, indirect next hop data that map the leaf nodes of the forwarding tree to the next hop data, wherein at least two of the leaf nodes of the forwarding tree contain indirect next hop data that references the next hop data for the same neighboring network device;

identify a key within a network packet;

traverse a subset of the nodes of the forwarding tree within a network device by testing at least one bit of the key per each of the traversed nodes, wherein values of the tested bits in the key determine a path traversed along the forwarding tree until reaching one of the leaf nodes of the forwarding tree;

upon reaching a leaf node of the traversed path, use the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree; and

forward the packet to the selected next hop.

Claim 29 (Previously presented): The computer-readable medium of claim 28, wherein the instructions cause the processor to store the forwarding tree as a radix tree.

Claim 30 (Previously presented): The computer-readable medium of claim 29, wherein the instructions cause the processor to store the indirect next hop data as a respective data pointer within each of the leaf nodes,

wherein the data pointer within each of the leaf nodes reference the next hop data external to the forwarding tree.

Claim 31 (Original): The computer-readable medium of claim 28, wherein the instructions cause the processor to store an array of next hop data elements, and further wherein the portion of the next hop data comprises at least one next hop data elements.

Claim 32 (Previously presented): The computer-readable medium of claim 28, the instructions cause the processor to:

receive a packet comprising network update information; and

modify the next hop data in response to the network update information.

Claim 33 (Previously presented): The computer-readable medium of claim 28, the instructions cause the processor to:

store routing information within a routing engine, wherein the routing information represents routes within a network; and

store the forwarding tree, the indirect next hop data and the next hop data within a packet forwarding engine.

Claim 34 (Previously presented): The computer-readable medium of claim 33, the instructions cause the processor to:

receive a packet comprising network topology update information;

update the routing information within the routing engine; and

issue a message from the routing engine to direct the packet forwarding engine to modify the next hop data external to the forwarding tree in response to the network update information without modifying the forwarding tree.

Claim 35 (Previously presented): The computer-readable medium of claim 33, wherein the instructions cause the processor to store a copy of the forwarding tree, the indirect next hop data and the next hop data stored within the packet forwarding engine.

Claim 36 (Previously presented): The computer-readable medium of claim 33, wherein the instructions cause the processor to store a copy of the forwarding tree, the indirect next hop data and the next hop data stored within the packet forwarding engine, and issuing the message comprises analyzing the copy to identify the next hop for modification.

Claim 37 (Previously presented): A method comprising routing packets within a network using indirect next hop data that maps leaf nodes of a forwarding tree to next hop data stored external to the forwarding tree,

wherein the leaf nodes correspond to destinations within a computer network,

wherein the next hop data represents next hops within a network, and

wherein at least two different ones of the leaf nodes of the forwarding tree contain indirect next hop data that references a same next hop within the next hop data.

Claim 38 (Cancelled).

Claim 39 (Previously presented): The method of claim 37, further comprising storing the indirect next hop data within the leaf nodes as pointers to primary next hops and pointers to backup next hops.

Claim 40 (Previously presented): The method of claim 37, wherein the forwarding tree comprises a radix tree.

Claim 41 (Previously presented): The method of claim 37, further comprising:

receiving a packet comprising network update information; and

modifying the next hop data in response to the network update information.

Claim 42 (Original): The method of claim 37, further comprising storing the indirect next hop data within a packet forwarding engine.

Claim 43 (Previously presented): The method of claim 42, further comprising:

receiving a packet comprising network topology update information;

issuing a message from a routing engine to direct the packet forwarding engine to modify the next hop data in response to the network update information.

Claim 44 (Original): The method of claim 42, further comprising storing a copy of the indirect next hop data within a routing engine.

EVIDENCE APPENDIX:

NONE

RELATED PROCEEDINGS APPENDIX:

NONE